

**Mr. Giansante**



# **Visual Basic**

**Slide Puzzle**

**August 2016**

# Slide Puzzle

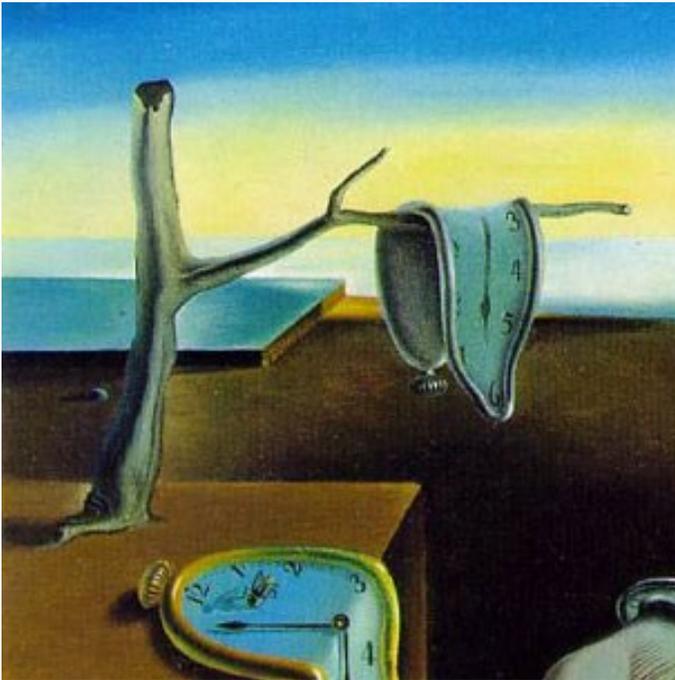
## Difficulty Level:

Beginner  
Intermediate  
Advanced

## Help References:

Button Control  
MenuStrip Control  
ImageList Control  
Random Numbers

Write a Visual Basic program that simulates a Sliding Puzzle. Your puzzle must have sixteen pieces.



This image is part of the painting "La Persistencia de la Memoria" created in 1931 by Salvador Dalí.

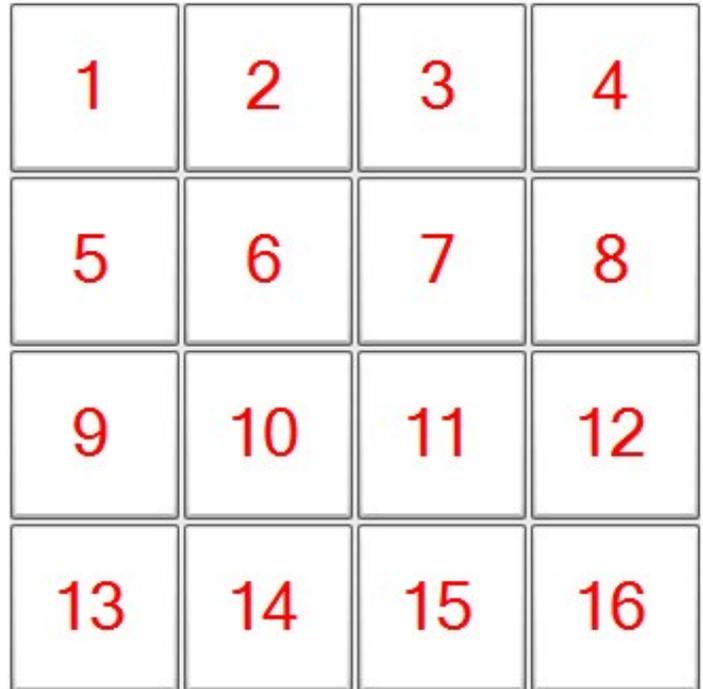
## Setting up the Playing Grid

The playing grid will consist of 16 Button controls.

Create one Button control and set the BackColor property to "White" and clear the Text property.

Use the Size property to make the button a perfect square. Dali images are 75 by 75 pixels (hence Button controls should be approx. 82 by 82 pixels)

Once the Button control is ready, copy it and paste 15 times to form the playing grid.



## Using an ImageList Control

The individual puzzle pieces will be stored in an ImageList Control.

From the Toolbox, insert an "ImageList" Control.

Set the "ColorDepth" property to "Depth24Bit".  
Set the "ImageSize" property to ...  
"75, 75" if you are using the Dali images

Select the "Images" property.

Click on the "..." button next to the word "(Collection)"

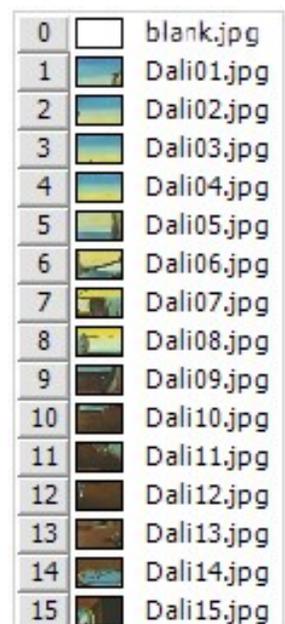
Select all the tiles (including the blank one).

Click on the "Add" button.

Re-order the images in the ImageList Collection if necessary.

### Important:

Set the **ImageList** property of each of the 16 Buttons to **ImageList1**.



# Slide Puzzle

---

## Declaring the Variables

In General | Declarations ...

```
Dim r As New Random
Dim rand(16) As Integer
```

## Randomizing the Puzzle Board

The following code goes in the "New Game" button ...

**Note:** This method is slightly more sophisticated than the method used previously in the Cell Phone Game.

```
'clear the current playing grid
For i = 1 To 16
    rand(i) = -1
Next i

'shuffle the pieces on the playing grid
For i = 0 To 15
    Do
        x = r.Next(1, 17)
    Loop Until rand(x) = -1

    rand(x) = i
Next i

Button1.ImageIndex = rand(1)
Button2.ImageIndex = rand(2)
Button3.ImageIndex = rand(3)
Button4.ImageIndex = rand(4)
Button5.ImageIndex = rand(5)
Button6.ImageIndex = rand(6)
Button7.ImageIndex = rand(7)
Button8.ImageIndex = rand(8)
Button9.ImageIndex = rand(9)
Button10.ImageIndex = rand(10)
Button11.ImageIndex = rand(11)
Button12.ImageIndex = rand(12)
Button13.ImageIndex = rand(13)
Button14.ImageIndex = rand(14)
Button15.ImageIndex = rand(15)
Button16.ImageIndex = rand(16)
```

# Slide Puzzle

---

## Coding the Buttons

Code for the Button control in the top lefthand corner ...

If the Button in the top lefthand corner (ie. position 1) is clicked, we much check for a blank space in position 2 (to the right of the Button) and position 4 (below the Button).

```
If Button2.ImageIndex = 0 Then
    Button2.ImageIndex = Button1.ImageIndex
    Button1.ImageIndex = 0
End If
```

```
If Button5.ImageIndex = 0 Then
    Button5.ImageIndex = Button1.ImageIndex
    Button1.ImageIndex = 0
End If
```

checkforwin() ←————— **Note:** You will write this procedure in the next step.

All 15 other Buttons will have similar code. Some buttons have 2 locations to check, some have 3, and some have 4.

For example, for the Button in position 6, you must check for a blank space in position 2 (above the Button), position 5 (to the left of the Button), position 7 (to the right of the Button) and position 10 (below the Button).

## Checking for a Win

Create a new procedure called checkforwin()

**Hint:** If Button1.ImageIndex = 1 And Button2.ImageIndex = 2 And ...

## The "Cheat" Button

Your game should provide a "Cheat" Button or menu item (possibly under the "Help" menu).

This button should solve the puzzle completely except for one move.

**Hint:** Button1.ImageIndex = 1  
Button2.ImageIndex = 2

# Slide Puzzle

Please fill out all the information in this column using a pen before getting this assignment marked.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

\_\_\_\_\_  
Class

## Academic Honesty

The work I am submitting is completely my own creation and has not been copied from anyone else's work. If I have received help on this project, the names of those who have assisted are listed below.

\_\_\_\_\_  
Signature

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## Pre-Marking

The following people have pre-marked this assignment: (minimum of two)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## Deductions

Each  which is checked indicates the item was not satisfactory and results in the loss of one mark.

Deduction for late assignment is 1 mark per day.

Deduction for program crashing is 2 marks.

## User Interface

- Appropriate Text in Title Bar (Form.Text)
- Separators and Access Keys in Menu, Capitalization
- "Exit" DialogBox meets requirements
- "About" DialogBox meets requirements
- Efficient and Esthetically-Pleasing User Interface
- Appropriate controls are used
- Spelling and Grammar are correct

## Code / Programming Style

- Variables are declared and logically named
- Code is commented where appropriate
- Code is indented and spaced to show
- Code is efficient

## Work Ethic / Problem Solving Skills

- Problem Solving Skills are demonstrated
- Makes Productive Use of Time
- Only Seeks Help when Necessary

## Program-Specific Criteria

- "New Game" button works
- Solved puzzle can be displayed and hidden
- Each possible tile slide combination works
- Tiles cannot be slid into used space
- Message is displayed when puzzle is solved
- Cheat Button - Positions tiles in a way for easy win

## Comments

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Mark** \_\_\_\_\_ / 10